Mixing Loop Control using Reinforcement Learning

Overgaard, Anders; Kallesøe, Carsten; Bendtsen, Jan Dimon; Nielsen, Brian Kongsgaard

Published in:
E3S Web of Conferences

DOI (link to publication from Publisher):
10.1051/e3sconf/201911105013

Publication date:
2019

Document Version
Publisher's PDF, also known as Version of record

Mixing Loop Control using Reinforcement Learning

Anders Overgaard[1,2], *Carsten* Skovmose Kallesøe[1,2], *Jan* Dimon Bendtsen[2], and *Brian* Kongsgaard Nielsen[1]

[1]Grundfos Holding A/S, Control Technology, Bjerringbro, Denmark
[2]Aalborg University, Department of Electronic Systems, Section of Automation and Control, Aalborg, Denmark

Abstract. In hydronic heating systems, a mixing loop is used to control the temperature and pressure. The task of the mixing loop is to provide enough heat power for comfort while minimizing the cost of heating the building. Control strategies for mixing loops are often limited by the fact that they are installed in a wide range of different buildings and locations without being properly tuned. To solve this problem the reinforcement learning method known as Q-learning is investigated. To improve the convergence rate this paper introduces a Gaussian kernel backup method and a generic model for pre-simulation. The method is tested via high-fidelity simulation of different types of residential buildings located in Copenhagen. It is shown that the proposed method performs better than well tuned industrial controllers.

1 Introduction

In Europe buildings account for 40% of the total energy usage. In the residential sector space heating accounts for 66% of the building energy consumption[1]. It is predicted that scheduling and improved control can lead to savings of 11-16% [2]. This huge savings potential is the reason that building control keeps being an active research area, see reviews [3] and [4]. In this work the focus is on building heating via mixing loops. Mixing loops are used to ensure proper comfort, heat power utilization and energy savings in buildings. Low heat power utilization leads to low efficiency in the supply coming from district heating. It has been shown that lowering the return temperature by 10^oC gave a heat loss reduction of 9.2% and pump energy reduction by 56% at the district heating plant [5]. So why is this important for the end user? The district heating plants are starting to enforce proper heat water cooling through added fees on a high return temperature. Ensuring proper heat power utilization in the control of the mixing loop can therefore also help reduce the end costumers cost of heating the building.

A lot of research has been done on optimal building thermal control, often in the form of Model Predictive Control (MPC). Examples of this are [6] and [7]. Here large savings was shown by using an MPC compared to traditional control strategies. The disadvantage of MPC is the reliance on accurate models of the building, especially when the product is installed into many different buildings. Different methods for identifying models of the building using data for MPC has been been explored. In [8] artificial neural networks are used for building the model for MPC, while in [9] subspace methods are used. In this work an alternative approach for learning optimal control through data will be investigated by using reinforcement learning to control a mixing loop. The result in [10] show that reinforcement learning is competitive with an MPC on a power system even when a good model is available. Even though Reinforcement Learning has been around for a long time, recent results have increased its popularity. This attention is mainly brought on by the Reinforcement Learning algorithm AlphaGo's ability to learn, tabula rasa, how to beat the world champion of the game Go [11]. Reinforcement learning has also been tried on HVAC applications. In [12] reinforcement learning was used to control passive and active thermal storage. Simulated reinforcement learning was used where the controller is getting priori knowledge from simulation. The result in [13] showed savings in heat-pump thermostat control by using reinforcement learning. In [14] a batch reinforcement learning method was used to control a heat-pump.

In reinforcement learning the rate of convergence towards optimal control is an issue, since it often requires a lot of training. In this work a Gaussian kernel backup rule is suggested to improve initial convergence in tabular Q-learning. Kernel based methods have been used in reinforcement learning, but mostly in regards to function approximation methods such as in [15].

The paper starts with an introduction to Reinforcement Learning in Section 2. The concept of building heat supply via a mixing loop is provided in Section 3. In Section 4 the proposed method using Gaussian kernel backup in Q-learning is presented. Section 5 explains the simulation setup. The results are presented and discussed in Section 6. The paper ends with the concluding remarks in Section 7.

2 Preliminaries

In this section reinforcement learning will be introduced. For a more thorough description see [16]. In Fig. 1 is a general reinforcement learning setup where an agent interacts with an environment.
The environment is in a state S_t at time t. "States" is here meant as all the information the agent receives about the environment. The environment also sends out a reward

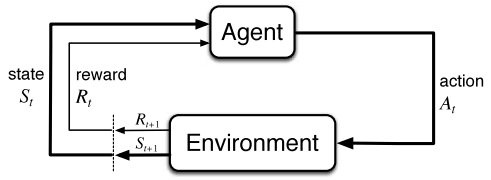

Fig. 1. Agent-environment interaction [16].

determining the instantaneous value of being in this state. The agent seeks to maximize the cumulative reward called the return [16]

$$G \doteq \sum_{k=0}^{T-t-1} \gamma^k R_{t+k+1} \tag{1}$$

where γ is the discount factor that lies in the interval $0 \leq \gamma \leq 1$. A higher discount factor will cause the agent to strive for longer term return, but will also increase the convergence rate of the learning agent. T is the final time step. For episodic task this is the end time, but for continuing tasks $T = \infty$. Having both $\gamma = 1$ and $T = \infty$ is not feasible as this would lead to infinite return.

The agent uses a policy, π_t, which goal is to maximize the return. The policy maps the states to an action, hence it is similar to a control law. The mapping can be of stochastic nature or deterministic.

The next element of Reinforcement learning is the value function [16]

$$V_\pi(s) \doteq \mathbb{E}\left[G_t | S_t = s\right] \tag{2}$$

The function describes that if starting in state s and continuing to follow policy π, the expected return will be G_t.

By adding onto the value function we get the state-action value function

$$Q_\pi(s,a) \doteq \mathbb{E}\left[G_t | S_t = s, A_t = a\right] \tag{3}$$

Which describes the expected return of being in state s, taking action a and afterwards follow policy π.

The goal in reinforcement learning is to find the optimal policy. This is often done through policy iteration by alternating between evaluating V_π using π and improving π using V_π. A greedy policy is a policy that always chooses the action which yields the highest return and is defined as

$$\pi_g(s) \doteq \arg\max_a q(s,a) \tag{4}$$

Such a policy fully exploits the current state-action value function, but the downside is that it does not explore and perhaps updates the state-action value function in such a way that the policy can be improved. This is the recurring problem of exploitation versus exploration. Proofs of convergence towards optimality often relies on exploration for reinforcement learning methods. So both exploitation and exploration needs to be done. A simple way to achieve that is the ε-greedy method. Here the greedy action is chosen with probability $1-\varepsilon$ and the rest of the times a random action is taken to explore.

The last element introduced is the learning rate, α, chosen from $0 < \alpha \leq 1$. This determines how much the newly learned information will override older information when updating the value function. In an environment that is fully deterministic the best learning rate is simply 1. Introducing stochastic behaviour such as noise or disturbances not contained in the states changes this towards supporting a lower learning rate.

3 Building Heat Supply via Mixing Loop

The Mixing Loop application is here described in short. This is done to get an understanding of the system, which is necessary for describing a reward function and choosing states and actions for the Q-learning. A simple model of the application is here described by a building with only one zone with one radiator as seen in Fig. 2.

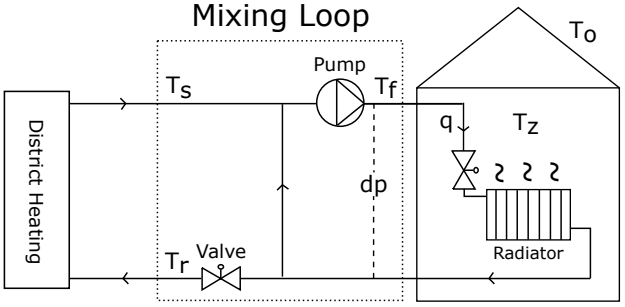

Fig. 2. Simple schematic of mixing loop application

The zone temperature is controlled by a thermostatic valve. The heat power is supplied via a mixing loop from district heating. The change in zone temperature is here described as the difference between heating, load and disturbance powers

$$C_z \dot{T}_z = \Phi_h + \Phi_L + \Phi_d, \tag{5}$$

where C_z is the heat capacity of the zone and T_z is the zone temperature. $\Phi_{h/L/d}$ are the heating, load and disturbance powers. The load power is the cooling acting on the zone from outside the building envelope. The disturbance power is all the remaining power acting on the system, also referred to as free heat. The majority of the disturbance power is created by the occupancy of the house and electric appliances.

The heat power is supplied by a radiator, which is here described as

$$\Phi_h = C_r \left(\frac{T_f + T_r}{2} - T_z \right)^n, \tag{6}$$

where C_r is the thermal conductance of the radiator, T_f is the forward water temperature, T_r is the return water temperature and n is a radiator constant. The heat power can also be described via the heating water as

$$\Phi_h = c_w q \left(T_f - T_r \right), \tag{7}$$

where c_w is the volumetric heat capacity of the heating water and q is the volume flow rate.

Via these two equations for heat power, the return temperature can be solved for, whereby the dependencies for heat power are

$$\Phi_h = f\left(q, T_f, T_z\right). \tag{8}$$

The flow rate

$$q = g(u)\sqrt{\Delta p}, \tag{9}$$

is a function of the thermostatic valve's opening degree u and the differential pressure Δp.

Typically a P controller determines the opening degree of the valve

$$u = K_p \left(T_{ref} - T_z \right), \tag{10}$$

Where K_p is the proportional gain and T_{ref} is the reference temperature set by the user.

The mixing loop controls T_f and Δp. By opening the control valve and mixing hot water at temperature T_s with return water having temperature T_r in a ratio that gives the desired T_f, see Fig. 2. Δp is controlled solely by the pump speed since the mixing loop hydraulically decouples the zone from the supply. The objective is to supply enough heating power for the system to keep the reference temperatures. By looking at (8) and (9) it can be seen that while the thermostatic valve controls the heat power, T_f and dp influences the gain of the controller. This means that by controlling T_f and dp only the gain of the thermostatic control can be influenced, except for saturation situations which is what is utilized for setback. The objective providing enough heat power has to be kept without increasing the pump pressure too much or increasing the return temperature leading to energy losses in the heat distribution. By (5) knowing the load and disturbance power heat power could be controlled as $\Phi_h = \Phi_L + \Phi_d$. The caveat of controlling by balancing the heat load is that if any unaccounted disturbance happens the thermostatic valve will be in saturation and will not be able to reject the disturbance. In mixing loop control it is not desirable to control in ways that eliminates the thermostatic valve's disturbance rejection.

The reward defines the control objective. For heating systems two features are important to optimize: comfort and cost. However, these two features can be described in various ways. For cost it is chosen to include the cost for the pump power, and the cost for the heat power. Other costs that could be included could be the cost of wear and tear of components such as the pump, pipes and valves or commissioning time when installing the HVAC system, but these are not included in this work.

The pump power cost is calculated as

$$\psi_{pump} = \Phi_{pump} \Omega_e, \tag{11}$$

where ψ_{pump} is the pump power cost, Φ_{pump} is the pump power consumption and Ω_e is the price of electric power. In this work Ω_e is kept constant at $0.27€/kWh$. If e.g. load shift is desired this should of cause be changed to a time dependant price. In this work the heat source is district heating, where a high return temperature reduces the efficiency, mainly through added heat losses in the distribution network. District heating companies often penalize high return temperatures by increasing the heat power cost as a function of cooling of the heating water. The additional cost is added differently dependent on the district heating company. In this work it is done like the district heating company in Copenhagen, "HOFOR", implements this [17].

$$\psi_{heat} = \Phi_{heat} \Omega_{heat} \eta. \tag{12}$$

Here ψ_{heat} is the heat power cost, Φ_{heat} is the heat power used, Ω_{heat} is the base price of the heat power at $88.9€/kWh$, and η is a price correction for cooling of the heating medium calculated as

$$\eta = 1 - \left(\frac{1}{125} (T_s - T_r) + \frac{33}{125} \right) \tag{13}$$

This means that the price of the heat power increases 0.8% per ^oC that the cooling of the heating medium is lower than $\Delta 33^oC$.

The heat comfort can be measured in different ways; here, the highest zone temperature error is used

$$e_{max}(t) = \max_{i \in \{1, \cdots, n_z\}} \left| T_{ref} - T_{z,i}(t) \right|, \tag{14}$$

where n_z is the number of zones. This ensures the lowest maximum error. Other ways of describing comfort can be the number of times temperatures exceeds a given bound. In this work only night setback is used, but other setback periods can be used via calendar functions or leaning patterns of the inhabitants. The difficult part about night setback is that it is dependent on the specific building. Both how much and for how long the temperature can be changed while ensuring comfort when setback ends varies. Not only from building to building, but also as a function of other states such as outside temperature. When reheating after a setback an optimal reheat "speed" is also important otherwise high return temperature will be imposed due to high flows and forward temperature. High return temperature during reheating is costly since a high amount of heat power is being consumed. Doing this in an optimal fashion should be learned by the reinforcement learning agent.

4 Q-learning with Gaussian Kernel Backup

4.1 Q-learning

The reinforcement learning method used here is Q-learning. Q-learning was first described in [18] and is defined by the backup

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \left[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t) \right] \tag{15}$$

A strength of Q-learning is that it directly finds the value of taking an action in a given state and afterwards following an optimal policy. This makes it model-free as no transition model of the environment is needed. A requirement for convergence towards the optimal policy is that all state-action pairs continue to be visited and updated. The formal proof of convergence can be found in [19]. To ensure convergence the ε-greedy method is used with $\varepsilon = 0.1$.

Q-learning is here single-step, as seen by the term $[\max_a Q(S_{t+1}, a)]$, but can be extended to multiple steps. The learning rate α is set to 0.2 and the discount rate to 0.4.

In this work a tabular version of Q-learning is used to ensure convergence. This is feasible when keeping a low dimensionality of the state-action space, $\mathbf{Q}$. The state action space used can be seen in Table I.

4.2 Choosing Reward

The reward function of a mixing loop is a multi goal reward system where it seeks to supply the best heat comfort for the building while minimizing cost. When it is deemed that setback can be used, the heat comfort goal vanishes and only the cost remains. The cost that the agent should minimize is the combined cost of the heat and pump power. Due to this multi objective reward a weighting factor, β, is needed, which determines the scaling between improving heat comfort and minimizing cost. In this work $\beta = 0.5$ unless otherwise stated. The reward then becomes

$$R(t) = \begin{cases} -(e_{max}(t)^2 + \beta(\psi_{heat}(t) + \psi_{pump}(t))) & 6 \leq t \ mod(24h) \leq 21 \\ -\beta(\psi_{heat}(t) + \psi_{pump}(t)) & otherwise \end{cases}$$

Here e_{max} is the maximum temperature error out of all the zones, squared to punish larger errors harder. Heat power cost ψ_{heat} and pump power cost ψ_{pump} was described in (12) and (11). Additionally a soft constraint is added such that low reward is given if any zone temperature goes below 16^oC.

Recall that the reinforcement learning seeks to maximize the cumulative reward. This ensures that an action that decreases power and therefore increases the reward during setback is only good if the building can reach the heat comfort giving high reward when setback is off.

4.3 Choosing States and Actions

As seen in section 3 there are a lot of states that would give added information for the agent. However in this work the focus is on making the minimal state-action space due to working with tabular methods where the state-action space and therefore learning rate suffers greatly from the curse of dimensionality. Another reason for keeping the state space small is the sensors needed for the information. Choosing the states is done by the definition given by [20]:

A state variable is the minimally dimensioned function of history that is necessary and sufficient to compute the decision function, the transition function, and the contribution (here the reward) function.

This selection is here done from the knowledge of the application, but could also have been done via correlation investigation.

States	Size of dimension	Range
Outside Temperature	21	-20 to 20 [^oC]
Time of day	24	1 to 24 [hours]
Actions	**Dimension**	**Range**
Pump Diff. Pressure	5	0 to 0.4 [bar]
Forward Temperature	31	15 to 75 [^oC]

TABLE I
STATE-ACTION SPACE

To ensure the zone temperatures enough heat power should be available for the thermostats. The needed heat power is a product of the load and the free heat, where the load is given by the outside temperature and the free heat given by multiple factors. Due to this T_o was chosen as a state. The free heat is not added explicitly in states in this work to reduce dimensionality, but later work could explore inclusion of indicators such as number of inhabitants present, solar radiation or electric appliances. Time is added as a state as $R(t)$ depends on it. Furthermore time of day can also capture periodic phenomenons, for example if free heat contains daily patterns.

The actions for the mixing loop application are the forward temperature and differential pressure, see section 3. Due to the nature of pumps the pressure is limited at higher flows. In the situation where the set point from the controlling agent is higher than the pump can supply it is set to max. The minimum forward temperature is 15^oC however due to the nature of a mixing the lowest forward temperature that can be supplied is the same as the return temperature at that given time. In the same way the maximum temperature is only as high as the supply temperature which in this case is controlled to 75^oC. So when choosing a forward temperature the agent can only choose from $T_r(t) <= T_f(t) <= T_s(t)$.

4.4 Gaussian Kernel Backup and pre-simulation

In tabular reinforcement learning using the Q-learning backup rule, the situation can occur where one specific state-action pair has been visited multiple times, but one in vicinity has never been explored. In this case there would be no knowledge of the state in the immediate vicinity since it has never been visited. Due to the priori knowledge of the "smoothness" of the application there must be knowledge to be gained about the optimal action in S_2 from the knowledge about S_1. This comes naturally when using function approximations such as kernel-based methods, but not in the tabular case. To gain increased convergence rate a Gaussian kernel is therefore applied to the backup process. Instead of only doing backup of the one state-action pair, backup is done on all state-action pairs with decreased learning rate the further the state is from the visited state. The learning rates are distributed using a Gaussian kernel. First two indexing vectors are introduced. $\mathbf{x}_t$ is the vector describing the location in the state-action tabular $\mathbf{Q}(S,A)$ at time t. It contains the index for each dimension. $\mathbf{x}$ is the vector describing the location of the state-action pair that is being backlogged to. Both has the dimension $n \times x$, where n is the sum of states and actions, in this case 4.

Now the backup is done to all state-action pairs using the following backup rule

$$Q(\mathbf{x}) \leftarrow Q(\mathbf{x}) + \alpha K_\sigma(\mathbf{x}_t - \mathbf{x})[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t)] \tag{16}$$

Where K_σ is calculated using the Gaussian kernel

$$K_\sigma(\mathbf{x}_d) = \exp\left(-\frac{|\mathbf{x}_d|^2}{2\sigma^2}\right) \tag{17}$$

In this work $\sigma = 1$ and is lowered as time passes. As σ decreases the method will converge to classical Q-learning. In Fig. 3 an example of a surface between a state and an action in a trained Q state space with and without Gaussian kernel backup can be seen.

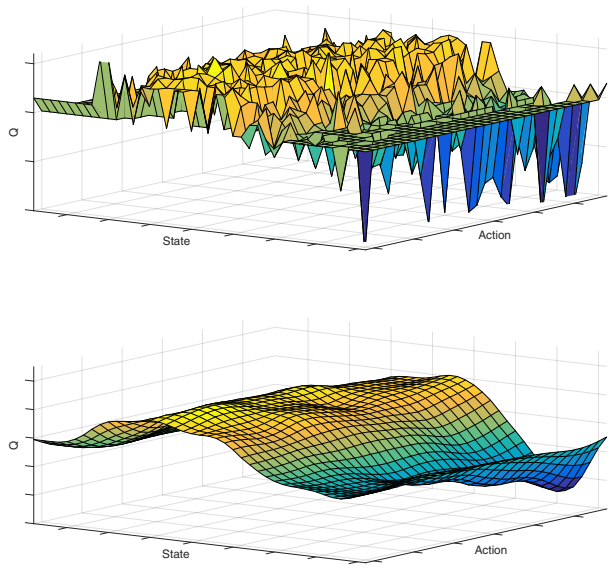

Fig. 3. Q surface compare without (left) and with (right) Gaussian kernel backup

Besides adding a Gaussian kernel pre-simulation is done to increase the initial performance of the controller. The pre-simulation is done via the generic model described in Eq. (5) to (10). The reason that a simple generic model is suitable for the initial guess, is that it should work for all the different buildings the product is installed to. The generic model was tried on the different buildings described in the next section and performed satisfactory. An example of this can be seen in the results Fig. 5.

5 Simulation Setup

The testing of the algorithm is done via simulation on high fidelity building models. The building model is made using the Modelica library "Buildings" [21]. To show the learning ability of the controller, it is used on three different buildings; House from 2015, house from 1960, apartment from 2015 and apartment from 1960. Fig. 4 shows the two floor plans of the house ($230m^2$) and the one of the apartment ($68m^2$).

Free heat from metabolism, electronics and hot water usage is modelled from typical daily, weekly and monthly patterns of usage. The difference between 2015 and 1960 buildings is the standard building materials of the time and standards for insulation, where Danish buildings from 2015 has a higher degree of insulation. Danish building code is used from each of the periods. The three buildings are situated in Copenhagen Denmark. For comparison some industrial standard controllers are used. There are typically four different tuning parameters to be chosen for the industrial controls. All buildings are supplied by 6 m head pumps. The industrial controllers are running proportional pressure. This means that the pressure rises proportional to the flow. The first parameter is the 3 different levels of proportional control that can be chosen on the selected pump. The next parameter is the outdoor temperature compensation. Here a saturated linear relation between outdoor - and forward temperature is often used. Besides this relation there is often a first order filter applied to the compensation

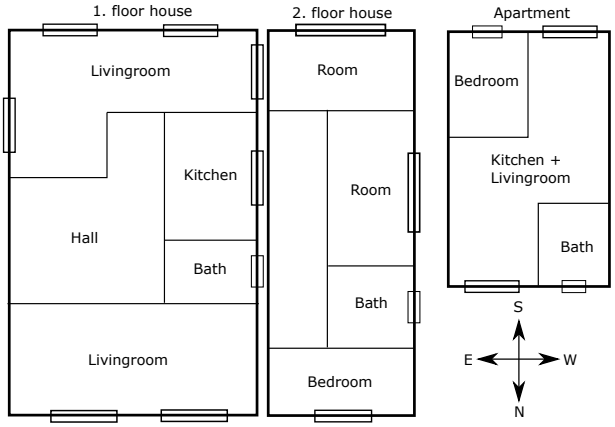

Fig. 4. Floor plans of house and apartment.

with a time constant, that is the third parameter. The time constant should be matched to compensate the dynamics of the building. If the compared industrial controller is without outdoor compensation then a notation of *NW* is used. The last parameter is a constant temperature that is to be subtracted from the outdoor compensated forward temperature during setback. Two setback temperatures are used; 15^oC and 30^oC and will be noted as such in the comparison tables. The pump curve, outdoor temperature compensation and filtering is tuned to the specific buildings to give a comparison against well tuned controllers. The tuned controller for modern house is C1, old house C2 and modern apartment C3. For all setback controllers, the setback period is between 9 p.m. and 6 a.m.

When comparing controllers the most important measure of the optimality of the controller is the returns, see (1). Normalized return is used which is the cumulative reward measured every 5 min. over the heating season. Here the heating season is chosen to be the 9 months September-May. For comparison of the controls the discount rate for this return is 1 meaning that all rewards during the heating season counts as equal. The return is normalized by the number of samples for readability. To also be able to compare the controllers directly on the comfort and cost two other measurements are given in the results, the Root Mean Square Error (RMSE) and the cumulative cost of running the system during the heating season.

6 Results & Discussion

In this section results showing the improvement of adding Gaussian kernel backup and pre-simulation will be shown. The results is a comparison with the industrial standard controllers. It is important to emphasize, when evaluating these results, that the industrial benchmark controller such as e.g. $C1-30$ has been carefully tuned for the specific house, which rarely is the case for real world buildings. This means that achieving performance as good as $C1-30$ via a self learning controller results in a much better performance than what is experienced in worse tuned buildings. In Fig. 5 the convergence of the reinforcement learning controller is shown for standard Q-learning backup, with Guassion Kernel backup, and finally adding pre-simulation. For each training duration, in interval of 1 month, the controller is run for a full heating season and the norm.

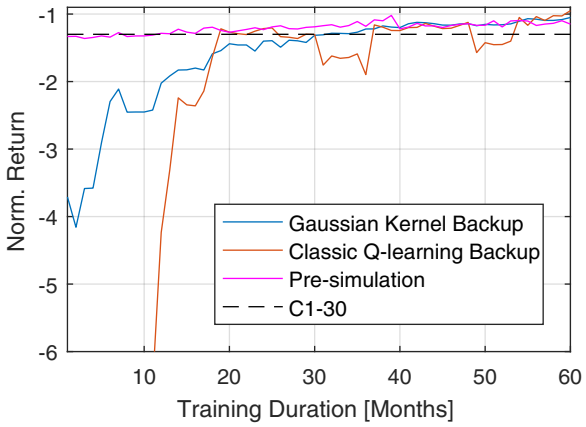

Fig. 5. Norm. Returns as a function of training duration.

return for that training duration is calculated. In this way it can be seen how the controller agent improves as a function of training duration. It can be seen that using the Gaussian kernel backup improves the initial performance until approximately the 18th month. Furthermore the Gaussian kernel backup improves the "stability" of the convergence, where the classic Q-learning deteriorates in periods, e.g. from 30-36 months. This graph also show the problem of learning Tabula Rasa. It takes around 30 months before reaching a satisfactory performing agent as the industrial controller C1-30, which is not feasible. Initialization using a priori knowledge by pre-simulating on the generic model provides a better initial controller. More work still needs to be done into increasing convergence rate, since training time still takes too long. The next results are comparisons of performance after 60 months.

In table II a comparison of the trained Q-learning agent with industrial standard controllers is shown. In parenthesis is the relative improvement the Q-learning provides compares with the industrial controller. The Q-learning agent manages to save energy in all scenarios. Only in two scenarios does the comfort decrease slightly, while gaining large savings. In the modern house the $C1-30$ is the best industrial controller measured in return. Compared to this the improvement in comfort and cost from using the Q-learning agent is 4.5% and 3.2%. Had the industrial controller been tuned worse for the modern house by choosing a setback constant of 15 the savings would instead be 8%.

Controller	Norm. Return	RMSE [^oC]	Cost €
Modern House - Copenhagen			
Q	-1.06	1.27	971
C1-15	-1.25	1.31 (3.1%)	1056 (8.0%)
C1-30	-1.19	1.33 (4.5%)	1003 (3.2%)
C1-30-NW	-1.29	1.39 (8.6%)	1018 (4.6%)
Old House - Copenhagen			
Q	-0.96	1.12	1920
C2-15	-1.25	1.11 (-0.9%)	2128 (9.8%)
C2-30	-3.24	1.20 (6.6%)	1985 (3.3%)
C2-30-NW	-4.13	1.26 (11.1%)	2022 (5.0%)
Modern Apartment - Copenhagen			
Q	-0.61	0.96	492
C3-15	-0.72	0.94 (-2.1%)	539 (8.7%)
C3-30	-0.74	0.96 (0.0%)	512 (3.9%)
C3-30-NW	-0.77	1.03 (6.8%)	521 (5.6%)

TABLE II
COMPARISON OF CONTROLLERS WITH SETBACK.

Fig. 6 shows an example of the time series data of one of the zone temperatures with the Q-learning agent and with the best tuned industrial controller $C1-30$ is shown. The Q-learning agent manages to increase energy savings by increasing the temperature reduction during setback. The Q-learning does this without violating comfort requirements by starting the reheating before leaving setback mode. If an increased comfort is desired the tuning parameter β in the reward function can be adjusted. To see how tuning β affects the performance, see Fig. 7. Here it is shown that the the agent with lower β starts to lower the temperature later to keep the comfort higher before setback occurs. Likewise it raises the temperature earlier before leaving setback to increase comfort. Recall that the agent is controlling forward temperatures and pressure, while a thermostat controls the zone temperature. It is by forcing the thermostat into saturation that the lowering of the zone temperature is possible from the mixing loop. Since the thermostat is a p-controller there will be a temperature error which is quite noticeable at around 5 o'clock in Fig. 7.

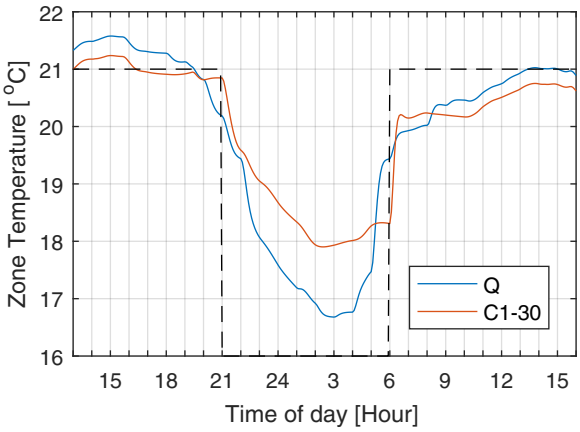

Fig. 6. Example of zone temperature during setback. Padded line is during setback the constraint and out of setback the set-point.

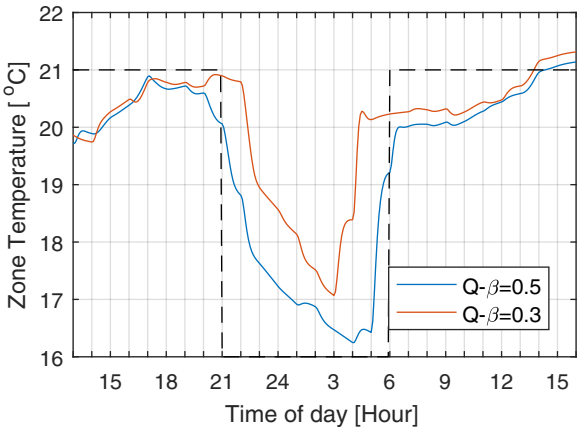

Fig. 7. Comparison of setback example with different β

If setback is disabled the Q-learning agent still manages to save cost while achieving comparable comfort compared

Modern House

Controller	Norm. Return	RMSE [^oC]	Cost €
Q	-2.01	1.23	1029
C1	-2.12	1.21	1076 (4.4%)

TABLE III
COMPARISON OF CONTROLLERS WITHOUT SETBACK

to the well tuned controller $C1$ in the modern house, which can be seen in table III. By comparing cost of the Q-learning agent with and without setback it can be seen that a saving of 5.6% is achieved through setback in the modern house. Comparing the industrial controller $C1$ without setback with the Q-learning agent with setback leads to 9.8% savings.

7 Conclusion

The motivation for this work is to investigate the performance of the reinforcement learning method Q-learning on building heating through mixing loops, while improving on the method by adding a Gaussian kernel backup and pre-simulation on a suggested generic model. In this work it was shown that even with the minimal information via a limited state-action space the reinforcement learning converges to a better performance than industrial standard controllers. Funnelling more information into the agent, such as free heat indicators, should increase the performance even further. However adding more information will decrease the convergence rate. To improve the convergence rate of Q-learning a Gaussian kernel backup method was added. Adding the Gaussian kernel added increased initial convergence rate, but even with the added convergence rate it still took 30 months to reach a satisfactory performance of the agent. By further adding pre-simulation on a generic model the initial controllers performance was greatly enhanced. The convergence rate however is still low, and need further improvement.

References

[1] L. Gynther, B. Lappillone, and K. Pollier, "Energy efficiency trends and policies in the household and tertiary sectors. An analysis based on the ODYSSEE and MURE databases," no. June, p. 97, 2015. [Online]. Available: http://www.odyssee-mure.eu/publications/br/energy-efficiency-trends-policies-buildings.pdf

[2] X. Cao, X. Dai, and J. Liu, "Building energy-consumption status worldwide and the state-of-the-art technologies for zero-energy buildings during the past decade," *Energy and Buildings*, vol. 128, pp. 198–213, 2016.

[3] P. H. Shaikh, N. B. M. Nor, P. Nallagownden, I. Elamvazuthi, and T. Ibrahim, "A review on optimized control systems for building energy and comfort management of smart sustainable buildings," *Renewable and Sustainable Energy Reviews*, vol. 34, pp. 409–429, 2014.

[4] A. I. Dounis and C. Caraiscos, "Advanced control systems engineering for energy and comfort management in a building environment-A review," *Renewable and Sustainable Energy Reviews*, vol. 13, no. 6-7, pp. 1246–1261, 2009.

[5] R. Sallent Cuadrado, "Return temperature influence of a district heating network on the CHP plant production costs," Ph.D. dissertation, 2009. [Online]. Available: http://hig.diva-portal.org/smash/get/diva2:228450/FULLTEXT01

[6] S. Prívara, J. Široký, L. Ferkl, and J. Cigler, "Model predictive control of a building heating system: The first experience," *Energy and Buildings*, vol. 43, no. 2-3, pp. 564–572, 2011.

[7] J. Široký, F. Oldewurtel, J. Cigler, and S. Prívara, "Experimental analysis of model predictive control for an energy efficient building heating system," *Applied Energy*, vol. 88, no. 9, pp. 3079–3087, 2011.

[8] A. Afram, F. Janabi-Sharifi, A. S. Fung, and K. Raahemifar, "Artificial neural network (ANN) based model predictive control (MPC) and optimization of HVAC systems: A state of the art review and case study of a residential HVAC system," *Energy and Buildings*, vol. 141, pp. 96–113, 2017.

[9] J. Cigler and S. Prívara, "Subspace identification and model predictive control for buildings," 2010, pp. 750–755.

[10] D. Ernst, M. Glavic, F. Capitanescu, and L. Wehenkel, "Reinforcement learning versus model predictive control: A comparison on a power system problem," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 39, no. 2, pp. 517–529, 2009.

[11] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. Van Den Driessche, T. Graepel, and D. Hassabis, "Mastering the game of Go without human knowledge," *Nature*, vol. 550, no. 7676, pp. 354–359, 2017.

[12] S. Liu and G. P. Henze, "Experimental analysis of simulated reinforcement learning control for active and passive building thermal storage inventory: Part 1. Theoretical foundation," *Energy and Buildings*, vol. 38, no. 2, pp. 142–147, 2006.

[13] D. Urieli and P. Stone, "A Learning Agent for Heat-Pump Thermostat Control," no. May, 2013, pp. 1093–1100. [Online]. Available: http://dl.acm.org/citation.cfm?id=2484920.2485092

[14] F. Ruelens, S. Iacovella, B. J. Claessens, and R. Belmans, "Learning agent for a heat-pump thermostat with a set-back strategy using model-free reinforcement learning," *Energies*, vol. 8, no. 8, pp. 8300–8318, 2015.

[15] D. Ormoneit and Å. Sen, "Kernel-based reinforcement learning," *Machine Learning*, vol. 49, no. 2-3, pp. 161–178, 2002.

[16] R. Sutton and A. Barto, "Reinforcement Learning: An Introduction," *IEEE Transactions on Neural Networks*, vol. 9, no. 5, pp. 1054–1054, 1998. [Online]. Available: http://ieeexplore.ieee.org/document/712192/

[17] HOFOR, "District Heating Prices." [Online]. Available: http://www.hofor.dk/fjernvarme/prisen-paa-fjernvarme-2017/

[18] C. J. C. H. Watkins, "Learning From Delayed Rewards," Ph.D. thesis, Cambridge University, 1989.

[19] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, no. 3-4, pp. 279–292, 1992. [Online]. Available: http://link.springer.com/10.1007/BF00992698

[20] W. B. Powell, *Approximate Dynamic Programming: Solving the Curses of Dimensionality: Second Edition*, 2011.

[21] M. Wetter, M. Bonvini, T. Nouidui, W. Tian, and W. Zuo, "Modelica buildings library 2.0," *14th International Conference of IBPSA - Building Simulation 2015, BS 2015, Conference Proceedings*, vol. 7, pp. 253–270, 2015.